

NAVIGATION, CONTROL, AND PARAMETER IDENTIFICATION OF AN UNMANNED SUBMERSIBLE

C. D. Ozimina
G. J. Bierman

Naval Research Laboratory, Washington, DC
Factorized Estimation Applications, Inc., Canoga Park, CA

ABSTRACT

This paper provides an overview of a new autonomous submersible, its existing navigation and control algorithms, and a proposed parameter-adaptive control structure. Design details of an extended Kalman filter, to be used for both state estimation and explicit parameter identification, are presented.

INTRODUCTION

The Naval Research Laboratory (NRL) is presently conducting research in the area of unmanned and untethered submersibles. A broad objective of this research effort is to develop the technology base and design procedures for a new class of autonomous free swimming submersibles.

NRL has previously designed and currently operates a towed sensor platform [1] which is used for general oceanographic work. In a typical research mission the towed vehicle is used to survey the ocean floor or to collect environmental data at various depths. Due to increased ship-board operating costs, personnel reductions and changing oceanographic research requirements, it is desirable to augment or replace the towed vehicles with an unmanned free swimming submersible. An unmanned and free swimming replacement, however, must be capable of automatic navigation, guidance and control with a minimum of external communication. Hence, the autonomous submersible design, unlike the towed platform, must contain its own navigation and control equipment. It also should possess reasonable speed and endurance for maximum effectiveness, within the limitations of a low cost power source (e.g., lead-acid batteries). Finally, the cost of the submersible is an important consideration.

In the first phase of submersible vehicle technology development at NRL, an unmanned free swimming submersible (UFSS) has been constructed and partially tested. An interesting feature of the UFSS is its microprocessor based guidance and control system, which provides for autonomous operation. A minimum-cost, minimum-risk approach was taken to develop the first phase guidance and control software algorithms and hardware. This approach lead to the development of classical,

fixed gain analog control equations, which were digitized for implementation in the microcomputer.

One of the objectives in the second phase of unmanned submersible research is to conduct research leading to the development of advanced concept, microprocessor based, guidance and control algorithms. A candidate concept under consideration is an adaptive control algorithm, based on modern control and estimation theory techniques. These techniques employ Kalman filter state estimation and near real-time parameter identification, together with a linear quadratic controller having feedback gains based on current plant parameter estimates. An important factor in this adaptive control concept is the utilization of numerically stable and computationally efficient algorithms that can be implemented in a microprocessor type computer. The candidate approach we propose is based on Bierman's covariance factorization methods [12]. His factorization algorithms compensate for microprocessor wordlength limitations so as to maximize the filter's computational efficiency, while at the same time safeguarding against the dangers of numerical accuracy degradation and algorithm instability.

In this paper we provide an overview of the NRL research submersible, its present classical guidance and control system, and the proposed adaptive control structure. The major contribution of this paper is our parameter identification process which plays a key part in the adaptive control structure. The identification process selected is related to the extended Kalman filter, which combines both state and parameter estimation in a single algorithm.

RESEARCH VEHICLE OVERVIEW

A key feature of the UFSS is its specially shaped hull (Fig. 1), which is designed to minimize drag per unit volume at the design speed. The NRL designed shape extends the laminar flow region to approximately two-thirds of the hull length [2]. Theoretically drag computations, which have been verified by recent scale model tests (unpublished), indicate that the drag coefficient is approximately one-third that of the more conventional, torpedo-type, parallel cylindrical body. The UFSS hull is fabricated in two

U.S. Government work not protected by U.S. copyright.

sections: a fiberglass forebody (laminar flow region), and an aluminum afterbody. The hull length is approximately 20 feet with a maximum diameter of 4 ft and a volume of 125 cubic ft. Design speed is 5 knots and theoretical maximum speed at reduced drag conditions (laminar flow) is over 9 knots. Operating depth is 25 to 1500 ft and duration is approximately 24 hours at the design speed.

The vehicle is stabilized and controlled by aft-mounted vertical and horizontal fins with movable, trailing-edge control surfaces. The control surfaces, which we refer to as rudders (upper and lower) and elevators (port and starboard) are independently controlled for three-axis attitude control capability. Other active controls include variable buoyancy and mercury trim systems. Buoyancy is normally positive for safety reasons. The trim system is used to adjust the longitudinal axis CG (center-of-gravity) and CB (center-of-buoyancy) displacement. The vertical axis CB-CG displacement is fixed at 0.5 inches for roll stability. The vehicle energy source consists of lead-acid batteries, which supply power to the propulsion system and control equipment. The propulsion system consists of a 1/2 hp, three-phase induction motor, which drives a three-bladed propeller at 0 to 200 rpm. An inverter-controller provides variable ac power to the induction motor for speed control.

The UFSS hull contains a pressure cylinder, which houses the batteries, electronics and other environmentally-sensitive equipment. The electronics package was designed and fabricated by NRL with commercially available equipment. Figure 2 provides an overview of the UFSS control system structure and equipment, which we refer to as the CCS (command and control system). The heart of the CCS is its microcomputer, which is based on an 8-bit microprocessor chip (Intel 8080A). Details of the microcomputer hardware design are shown in Fig. 3. As shown, the microcomputer contains 24K bytes of EPROM (programmable erasable, read only) and 5K bytes of RAM (volatile and read/write) memory. Digital I/O (input-output) consists of two serial data interfaces (RS-232 and current loop type) and 96 parallel digital lines, which are divided into ports of 8 lines each. Analog I/O consists of a 16 channel (32 single ended inputs) A/D (analog/digital) converter and a 8 channel D/A (digital/analog) converter. A teletype, which is connected to the microcomputer by an umbilical, is used to communicate with the CCS microcomputer prior to and upon completion of untethered operation. An acoustic subsystem, which consists of a transponder and command decoder at the vehicle end, is used to communicate with the UFSS during untethered operation. Additional details of the equipment in Fig. 2 are given in [3].

UFSS GUIDANCE AND CONTROL

A description of the existing (Phase 1) navigation, guidance and control algorithms follows.

Navigation and Guidance

Horizontal plane navigation is accomplished by a dead-reckoning algorithm which uses the speed and heading sensor inputs. Vehicle position, which is computed in geodetic (latitude-longitude) coordinates, is updated by periodic fixes from the OMEGA sensor. Note that the UFSS must "pop up" to near surface for an OMEGA fix due to in-water attenuation of the OMEGA signals. Horizontal (heading) and vertical (depth) plane guidance commands, which navigate the vehicle between waypoints, are computed by comparing estimated present position with desired position. The waypoints are a set of predetermined geodetic coordinates which are contained in a mission control table. If waypoint navigation is not desired, the guidance routine can be instructed to obtain time scheduled heading and depth commands from the mission control table. The mission control table is preprogrammed via teletype inputs prior to umbilical disconnect.

Heading Control

A functional block diagram of the heading control mechanization is shown in Fig. 4. As shown, the vehicle heading is controlled by an outer heading to rudder feedback loop with an inner yaw rate feedback loop for added damping. The basic control form shown in the figure was selected with the aid of classical analog control system design techniques (S-plane, root locus) and verified with the aid of a digital simulation. For nominal heading and yaw rate gains of -0.3 deg/deg and -0.3 deg/deg/sec, the vehicle heading response to a small signal step heading input is similar to a critically-damped, 2nd order dynamic system with a settling time (95%) of approximately 30 sec. The relatively low loop gains and the rudder command limit of 25 degrees provides satisfactory operation for heading changes up to 360 degrees. The vehicle's dynamics change, however, due to speed loss, vehicle turn rate limit (rudder saturation) and other nonlinear effects.

The real-time software for the heading control function was implemented by straight-forward discretization of the original analog design. A computation rate of 1 Hz, selected with the aid of the digital simulation, provides for near-analog operation with a duty cycle of only 7%. Arithmetic operations are in double precision (16 bit) and input-output data is 12 bits.

Depth/Pitch Control

A functional block diagram of the depth/pitch control design is shown in Fig. 5. As shown, the depth control is accomplished by an outer depth to elevator loop with integral compensation in the forward path. Inner pitch attitude, which is nearly proportional to depth rate, and pitch rate to elevator feedback loops provide for better closed loop dynamic characteristics. The use of pitch attitude feedback also allows for control of maximum pitch angle in order to limit speed loss during depth changes.

The UFSS is normally operated with approximately 25 pounds of net positive (upward) buoyancy for safety and recovery reasons. Ordinarily, a loss of buoyancy is expected at depth due to temperature and pressure effects. Uncertain salinity levels, however, can cause buoyancy to vary either way. A negative (downward) steady-state angle of attack is required to generate the necessary negative lift component for level depth operation. The required negative pitch angle, which is equal to vehicle angle of attack is steady state (no currents), is automatically computed by the integral compensator (Fig. 5). This compensation converts the depth control loop from a type 0 (steady state depth error) to a type 1 (no steady state error) control system. The integral compensation, however, has a destabilizing effect on the depth/pitch control loop dynamic characteristics. This results in a vehicle depth response (to a small step depth change) which resembles that of an underdamped second order dynamic system. Special integral control logic and a prudent selection of gains minimize the destabilizing effect of the integral compensator.

The basic depth/pitch control laws and gains were selected with the aid of classical control system design techniques and the use of a digital computer simulation. Reference [4] provides additional design details. Figure 6 shows additional details of the depth/pitch control algorithm. The integral control logic maintains the integral compensator output at null if (1) the depth error exceeds a preset level (25 ft), or (2) if vehicle depth is less than 10 ft. This logic effectively reduces depth overshoot for large depth changes and during the initial dive. The depth/pitch controller gains are scheduled for transition and constant depth operation. In the transition mode, a large depth gain ($K_D = -1.5 \text{ deg/ft}$) is selected and the pitch control command limits are relaxed to $\pm 25 \text{ deg}$. This mode enables large elevator commands in response to the initial dive command, while still maintaining closed loop operation, in order to overcome surface effects. We note here that prior to the initial dive command, the elevators are commanded to 15 deg up (open loop command, not shown in Fig. 6) in order to maintain a tail down configuration for maximum surface speed. Upon exceeding 10 feet of depth, the depth/pitch controller reverts to the depth (normal) mode of operation. The normal mode depth gain is $-.3 \text{ deg/ft}$ and the reduced pitch command is $\pm 10 \text{ deg}$. Nominal pitch (K_0) and pitch rate (K_q) gains are -1 deg/deg and -1 deg/deg/sec , respectively for both modes. The integral compensator gain (K_I) is $.01 \text{ sec}^{-1}$, which indicates a depth transient settling time of 5 minutes. This set of gains in the depth mode minimizes vehicle angle of attack and pitch perturbations, reduces speed loss and maintains small signal, near linear operation for dives and rises to or from maximum operating depth.

As in the heading control algorithm, the depth/pitch control computation rate is 1 Hz. A rectangular integration algorithm is used to implement the single digital integrator. Compu-

tations are also in double precision and the estimated duty cycle is 14%.

Speed, Ballast and Trim Control

The remaining vehicular control functions are implemented in an open loop manner. Speed commands are stored in the previously described mission control table in the form of propeller RPM. Outputs are issued in the form of an analog voltage to the propulsion motor controller. Commands to the trim and ballast control subsystems are also stored in the mission control table. A closed loop ballast control system, which maintains a minimal positive buoyancy for safety has been designed and simulated, but has not yet been implemented in the software.

ADVANCED CONTROL SYSTEM

In phase II of the NRL submersible vehicle research effort, both the existing classical (phase I) depth/pitch and heading control algorithms will be upgraded for self-adaptive control. The resulting adaptive control algorithm would severely tax the existing 8-bit microcomputer. Hence, the microcomputer will be upgraded in terms of speed and accuracy by means of a newer 16-bit microprocessor chip with a hardware multiply capability.

The general structure (Fig. 7) of the candidate adaptive controllers is based on the self-tuning regulator concept. The self-tuning regulator, as described by Åström and associates [5], consists of three basic parts: a controller, a parameter identification process, and some adaptation process to adjust the controller gains from the identified parameters. In our version of the self-tuning regulator, the controllers are based on the linear-quadratic (LQ) state regulator technique [6,9]. A well-known and desirable feature of the LQ state regulator is its minimum ($\pm 60 \text{ degrees}$) phase and infinite gain margin property [7,8] with the use of full state feedback. Considering that the UFSS can best be represented by a stochastic model, we select a Kalman filter to provide an estimate of the true full state (x in Fig. 7). Following the usual practice, we refer to LQ state regulator with a Kalman filter state estimate as a LQG (linear-quadratic-Gaussian controller) [6].

The process of identifying the parameters of a dynamic system from its input-output data is a well-researched but not completely solved problem. There are a variety of possible parameter identification methods (cf. [8]), with some methods suitable only for off-line identification, while others are more amenable to on-line implementation. In our extended Kalman filter (EKF) method (cf. [10] or [11]), the parameters to be identified are arranged in vector form (p) and are augmented to the nominal state vector (x in Fig. 7) as additional states. The parameter-augmented state equations are nonlinear because the p and x - states appear multiplicatively in the original x state equations. Hence, combined

estimation of x and p is a nonlinear filtering problem. When the nonlinear, augmented state equations are (for example) continuously linearized about the current "best" augmented state estimate, the normal Kalman filter algorithm can be applied to estimate both x and p states. This modification to the normal Kalman filter is described as the EKF in [11, Sec. 8.3]. Our approach, described in the next section, differs somewhat from this normal EKF method.

Having selected candidate controller and parameter identification methods, the final process to be selected is that of the actual on-line controller gain computation. An on-line computation of the actual controller gains (G in Fig. 7), for the current parameters is not practical. This computation requires either a backwards in time solution of a first-order, nonlinear, matrix-Riccati differential equation, or the solution of an algebraic version of this same equation (cf. [19, Chp. 10] for details). A candidate gain computation method, that is being considered, is to schedule the LQ gains from a precomputed gain set, based on the current parameters. Pre-stored gains are computed for the range of expected parameter values. A basic assumption in the LQ gain computation is the use of a linear, time-invariant vehicle dynamics model. This assumption indicates that the adaptive controller can only adjust to reduce the effects of initially uncertain or very slowly time varying parameters. In view of the difficulty in developing a good initial math model for a laminar flow body such as the UFSS, we accept this limited-sense adaptive control.

Details of the LQ gain scheduling process are the subject of another publication. In the remainder of this paper, we describe our on-line approach to EKF state and parameter estimation.

KALMAN FILTER DEVELOPMENT

Key Features

A batch sequential filter model is assumed, in which the dynamic parameters to be identified are modeled as Markov and piecewise constant. At the same time the vehicle state is modeled as a continuous time process, an assumption that is consistent with the vehicle dynamics. These assumptions allow model flexibility, so that we are not chained to determining dynamic parameter constants representing a time varying situation. Incidentally, these assumptions also lead to significant computational economics. Most importantly this model is much less sensitive to the effects of wildpoint data, consequently reducing the problem of filter divergence that so often appear in typical "extended" Kalman filter formulations.

Our filter design employs the Bierman U-D covariance factorization, $C = UDU^T$ (cf. [12]). Algorithm recursions for time and measurement updating the U-D factors are used in place of the conventional Kalman filter covariance recursions.

The Bierman algorithms are computationally efficient and insensitive to numerical roundoff. These features are important for microprocessor onboard implementation.

Vehicle Models

The complete equations of motion for the NRL submersible contain at least 12 dynamic states and up to five control inputs. Following the usual practice, linearized reduced-order models were developed for use in the adaptive depth and heading controllers. These models, which describe the perturbations about some nominal trim conditions, are in state space form

$$\dot{x}(t) = A x(t) + B u(t) + \zeta(t) \quad (1)$$

where x and u are vectors containing the dynamic states and control inputs, A and B are dynamics and control input matrices, and ζ is a zero-mean, Gaussian white noise vector process with covariance intensity Q . A and B are assumed to contain some initially uncertain or slowly time-varying elements. To illustrate our proposed EKF algorithm for use in the depth controller, we define the following candidate state and control input vectors for motion limited to the vertical and pitch planes: $x^T = (\hat{\theta}, \dot{\hat{\theta}}, \theta, h, \dot{h})$, and $u = \delta_e$ where θ and h represent perturbations about the nominal pitch and depth states, and δ_e is the perturbation about the trim elevator control input. A is a sparse 5×5 matrix

where $A_{2,1}, A_{3,2}, A_{5,4}, A_{1,1}, A_{1,2},$
 $A_{1,3}, A_{4,1}, A_{4,2}, A_{4,3}, A_{4,4},$

are the non-zero elements that are

known, but to varying degrees of uncertainty. Similarly B is 5×1 with non-zero element $B_{1,1}$, which is the elevator effectiveness. We assume that the states $\hat{\theta}, \theta$, and h are directly observable, but noise-corrupted. This leads to an observation model of the form

$$z(t_k) = H x(t_k) + v(t_k) \quad (2)$$

where $z = (h, \theta, \dot{\theta})$, H is a sparse matrix with 3 non-zero elements (1's), v is a zero-mean, white noise process vector (representing the sensor noise effects) with covariance R , and t_k represents the discrete time example.

The role of the filter process noise model, ζ , in (1) is to compensate for: (a) intentional mismodeling (such as excluding higher order derivative terms from a Taylor series model); (b) complex and/or poorly understood physics (such as location related anomalies, ocean current effects, temperature dependent effects, etc.); (c) random phenomena (such as the presence of heavy sea growth concentrations, laminar to boundary layer tripping, etc.); and (d) to add uncertainty to compensate for computational roundoff and integration error. When good data, sufficient both in quality and quantity, are present one need not depend heavily on the dynamic model, and in such cases the assumption of large process noise uncertainty intensity leads

to an estimator that depends heavily on the local data and to a much lesser extent on details of the dynamic model structure.

Since one of our key objectives is to identify the imprecisely known dynamic parameters we want our estimates to be model dependent; therefore, it is important to reduce the dynamic noise intensities as much as possible. In that way we will be, loosely speaking, maximizing the dynamic parameter observability. We are constrained to retain the dynamic noise effects due to (a), (b), and (c) but by employing an accurate integration algorithm and a U-D covariance factorized filter algorithm we need not include additional noise effects to compensate for roundoff.

Parameter Model

Although the dynamic parameters are poorly observable they do not affect the dynamic stability of the estimator. To avoid dynamic behavior that might change erratically depending on the most current data we use a batch sequential piecewise constant model. We note that the modeling assumptions act to produce a piecewise constant estimate that is a weighted average and, as such, it is less sensitive to the effects of individual data points. An additional strength of our technique is that by using correlated noise to model the dynamic parameters we have a fading memory effect (i.e. the old dynamic parameter estimates are dewighted in such a way that the current estimates tend toward improving the local fit).

To illustrate our procedure, suppose that $A_{1,2}$ and $A_{1,3}$ in (1) are sufficiently uncertain and require identification for gain scheduling. We define the parameter vector $p(t)^T = (A_{1,2}(t), A_{1,3}(t))$, where the p_i elements of p are assumed to be piecewise constant over some time span (for example, 5 to 10 seconds). To reflect the uncertain variation from interval to interval the p_i levels are modeled as constants plus a colored noise correction, δp_i , e.g.,

$$\delta p_i(t_{j+1}) = m_i \delta p_i(t_j) + w_i$$

where $i = 1, 2, j = 0, 1, 2, \dots$

$$m_i = \exp(-t_{j+1} - t_j) / \tau_i$$

τ_i are the process correlation times and

$$w_i = w_i(t_{j+1} - t_j) \text{ is zero mean with variance}$$

$$q_{w2}(i) = \sigma_i^2 (1 - m_i^2)$$

with σ_i^2 as the expected a priori mean square deviation from the nominal.

This model choice is key to our design, and it merits discussion. Our idea is to estimate a dynamic state, $x(t)$, and a constant dynamic parameter correction vector $\delta p^T = (\delta p_1, \delta p_2)$ over

the interval $t_j \leq t < t_{j+1}$. The strategy is that these element corrections do not update the dynamics matrix until time t_{j+1} . Experience with related aerospace problems is that δp estimates based on all the data appearing in the time interval t_j to t_{j+1} are less sensitive to the presence of wildpoint measurements. By not updating the dynamics matrix after each measurement is processed, we avoid the danger of having a single undetected bad measurement corrupt the dynamics and possibly induce instability. An incidental, but not trivial, benefit accrued to this design strategy is that there is a major computational savings related to generation of the transition matrix, one of the most time consuming of the various filter calculations.

The colored noise model described is introduced to allow stage to stage variation of the dynamics parameter corrections. Two extreme cases that demonstrate the flexibility of this model are: (a) τ small or zero, in which case we solve for new corrections terms at each Δt stage, which are (essentially) independent of the corrections chosen in the previous intervals, and (b) τ large, in which case we compute corrections that are weighted so as not to change the nominal much from its previous updated value. For time constants between these extremes, corresponding results follow.

Discrete Time Batch Sequential Formulation

In consonance with our announced strategy regarding process noise, to include it only where needed to compensate for model inadequacies, we assume that only the highest derivatives of the state vector, x , are noise driven. It is convenient and realistic to assume that there is a fine grid time increment Dt over which the control u and process noise ζ of (1) are piecewise constant; viz. take $t_{j+1} - t_j = \Delta t$ and set $Dt = \Delta t/5$. Combining these control and noise model assumptions with the dynamic parameter identification model of (1), gives the continuous time filter model

$$\dot{x} = A(t, t_j) x(t) + Bu(t) + \Gamma w(t) \quad (3)$$

where

$$A(t, t_j) = A_j; t_j \leq t < t_{j+1} \quad (3a)$$

$$u(t) = \delta_e(i, j); t_{j,i-1} \leq t < t_{j,i} \quad (3b)$$

$$\Gamma^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (3c)$$

$$w(t) = \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} \quad (3d)$$

The w_i are piecewise constant random variables for $t_{j,i-1} \leq t < t_{j,i}$. The variances of $w_1(t)$ and $w_2(t)$ are q_0 and q_h , respectively and the subscript i in (3b) and (3d) ranges from $i = 1$ up to $i = \Delta t/Dt$.

The continuous time model is transformed to

discrete form for a number of reasons. Our particular motives are (a) that the input/output structure of the problem is essentially discrete, because of the role of the digital controllers and processors, and (b) efficient, numerically stable, reliable and flexible algorithms are available with which to attack the discrete time problem. To convert the continuous formulation to discrete form recall that over a subinterval over which A , u and w are constant ($t_{j,i-1} \leq t < t_{j,i}$) we have the analytical representation

$$x(t) = \Phi_x(t, T)x(T) + \Psi(t, T)(Bu(T) + \Gamma w(T)) \quad (4)$$

where $T = T_{j,i-1}$ and Φ_x and Ψ are transition matrices. There are a number of ways to compare Φ_x and Ψ ; but because of space limitations, we do not discuss them here.

Let us return attention to the title of this section and define our discrete time batch sequential extended Kalman filter model. At time t_j we have estimates of the pitch-depth vector, and of the dynamics parameters; call them \bar{x} and \bar{p} , respectively. A prediction model, one that is to hold over a so called "batch" interval, t_j to t_{j+1} , is as follows. To go from $t_j + kDt$ to $t_j + (k+1)Dt$ use

$$x_{k+1} = \Phi_x(j)x_k + \Psi(j)Bu_k \quad (\text{prediction model}) \quad (5)$$

where $\Phi_x(j)$ and $\Psi(j)$ are based on dynamics parameter values \bar{p} , $x_0 = \bar{x}$. The integration step is Dt and u_k are the local elevator control deflection values. The index k ranges from $k = 0$ up to $k = n_{Dt} - 1$, where $n_{Dt} = \Delta t/Dt$. At time t_{j+1} the "inner loop is complete, and the nominal values to be used for the next Δt prediction interval are determined.

Our extended Kalman filter model involves corrections to be nominal x_k of (5) and corrections to the dynamic parameters, p ; these corrections denoted as δx_k and δp , respectively obey the stochastic model.

$$\begin{pmatrix} \delta x \\ \delta p \end{pmatrix}_{k+1} = \begin{pmatrix} \Phi_x(j) & \Phi_{xp}(j,k) \\ 0 & I \end{pmatrix} \begin{pmatrix} \delta x \\ \delta p \end{pmatrix}_k + \begin{pmatrix} \Psi(j) \Gamma w_k \end{pmatrix} \quad (6)$$

with initial correction estimates

$$\begin{pmatrix} \delta x \\ \delta p \end{pmatrix}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (7)$$

Estimates of x and p are

$$\begin{aligned} \hat{x}_k &= x_k + \delta \hat{x}_k \\ \hat{p} &= \bar{p} + \delta \hat{p}_k \end{aligned} \quad (8)$$

At time t_j our best estimates of x and p are \bar{x} and \bar{p} , determined from the model and data up to time t_j ; the representation (8) explains the initialization defined in (7). The Φ_{xp} term appearing in (6) is given by

$$\Phi_{xp}(j,k) = \int_{t_k}^{t_{k+1}} \Phi_x(t_{k+1}, s) F_{xp}^T(s) ds \quad (9)$$

where

$$F_{xp}^T(s) = \begin{pmatrix} x_k(2) & 0 & 0 & 0 & 0 \\ x_k(3) & 0 & 0 & 0 & 0 \end{pmatrix} \Big|_{t=s} \quad (10)$$

(i.e the second and third components of x , $\dot{\theta}$ and θ at time t_k are evaluated at $t = s$. Applying a simple trapezoidal quadrature approximation to (9) we obtain

$$\begin{aligned} \Phi_{xp}(j,k) &\approx \frac{Dt}{2} \left[\begin{pmatrix} x_{k+1}(2) & x_{k+1}(3) \\ 0_4 & 0_4 \end{pmatrix} + \right. \\ &\left. \Phi_x(j) \begin{pmatrix} x_k(2) & x_k(3) \\ 0_4 & 0_4 \end{pmatrix} \right] \quad (11) \end{aligned}$$

with 0_4 being a four component zero vector.

At this point, the model definition is complete and, to tie together the development, we recapitulate the key items to show how the results match the title description. That we have a discrete time model formulation is clear from (4) and (5). The formulation is batch sequential in the sense that estimates of the dynamics coefficients, p , are obtained from the n_{Dt} data samples in the Δt interval t_j to t_{j+1} . We have an extended filter in that the estimates of the pitch and pitch rate ($x_k(3)$ and $x_k(2)$, respectively) are used in the $\Phi_{xp}(j,k)$ computation, (10). Thus, estimates of p and x are used to compute the linear filter model coefficients. Finally, we implement a Kalman filter because the statistics of the data and dynamics noise models are used to satisfy a minimum variance criteria.

Remark: The claim of satisfying a minimum variance criteria is only a formal statement, because of the nonlinearity (in p) of our model, (1). We are, however, minimizing a weighted least squares performance functional that is much like a maximum likelihood performance index.

FILTER DESIGN TIME AND MEASUREMENT ASPECTS AND OVERVIEW

Experience with covariance mechanized Kalman filters (cf. [13] for example) has shown them to be numerically unreliable, particularly in applications involving limited word-length, and where key parameters are observed only indirectly. In the last few years there has been considerable progress, and a noteworthy contribution to the

state-of-the-art are the algorithms developed by Bierman, (12)-(15), which employ U-D covariance matrix factorization; i.e., the filter covariance matrix C is decomposed into its U-D factors, $C = UDU^T$, where U is unit upper triangular and D is positive diagonal. Bierman's idea is to replace the numerically unstable covariance matrix recursions with U-D factor recursions. Using the U-D decomposition instead of the covariance matrix entails no extra memory and a negligible increase in computational requirements. The algorithms involved have been thoroughly tested and are FORTRAN coded [16] in a form that is readily transferred to a minicomputer or microprocessor. An outline and comments on the filter design follow.

The measurement update portion (Fig. 8) of our EKF is based on the Schmidt EKF, as described by Jazwinski [11, Sec. 8.3]. In our version we use U-D covariance factors in place of the estimated error covariance matrix in Schmidt's formulation. We begin by initializing the measurement update process at discrete time t with the current (nominal or time updated) estimates of the states (\hat{x} , \hat{p} , $\hat{\delta p}$) and covariance factors (U , D). The measurements z (depth, pitch, pitch rate sensor outputs) are tested for validity by comparing the residuals, Δz with the predicted error variances. If the residuals are accepted, the states and covariance factors are updated (\hat{x}^+ , \hat{p}^+ , $\hat{\delta p}^+$, U^+ , D^+). Otherwise, the state and covariance factors are not updated. We note that the measurements are sequentially processed.

In the time update process (Fig. 9) the inputs x , U , and D (current or measurement updated) are propagated in time from t to $t + \Delta t$, with δp held constant. Upon completing the normal time updates, the current time, t , is compared to the next parameter update time T (batch), where the T (batch) times are separated by Δt . If t is less than T (batch), we output our estimates \hat{x} , and U , D factors prior to the next time or measurement update cycle. At a T (batch) time, we correct our model dynamics and thereupon set $\delta p = 0$. Details of the filter implementation are given in (17).

CONCLUDING REMARKS

We have outlined an advanced control system for a new autonomous submersible which employs an EKF algorithm for combined state and parameter identification. Because of its microcomputer implementation, special attention has been placed on computational details. No results are included and many issues yet need to be resolved. An interim continuous/discrete EKF has been simulated on a ground-based computer using matrix handling subroutines and 64 bit double precision arithmetic. The basic EKF approach to parameter identification was checked with both simulated and actual test data. Up to three parameters have been simultaneously identified using the simulated test data. In Fig. 10 we demonstrate the off-line identification of two parameters from simulated test data. The initial parameter

estimates are zero and the measurement update is 1 Hz. We admit to using square wave control inputs to excite the vehicle dynamics, which facilitated the identification. More extensive simulation tests to corroborate our filter/identification design are in progress.

REFERENCES

1. Gennari, J. J. and Bridge, R. B., "Naval Research Laboratory Deep Ocean Search and Inspection System," NRL Report 1733, June 1974.
2. Hansen, R. J., and Czul, E. C., "Hull Design of a Low-Speed, Laminar Flow Submersible," 1977 Review, Naval Research Laboratory, pp. 32-33.
3. Ozimina, C. D., Shirley, D. E., and Carey, E. W., "A Microprocessor Application in an Unmanned Submersible," Proceedings of MIMI '80 Asilomar Conference, Jan. 30 - Feb. 1, '80.
4. Ozimina, C. D., "Design of a Control System for an Unmanned Submersible," Proceedings of SOUTHEASTCON '79, Region 3 Conference, April 1-4, 1979.
5. Åström, K. J., et al., "Theory and Application of Self-Tuning Regulators," Automatica, Vol. 13, 1977, pp. 457-476.
6. Athans, Michael, "The Role and Use of the Stochastic Linear-Quadratic-Gaussian Problem in Control System Design," IEEE Trans. of Automatic Control, Vol. AC-16, No. 6, pp. 529-552, December 1971.
7. Safanov, M. G., and Athans, M., "Gain and Phase Margins for Multiloop LQG Regulators," IEEE Trans. on Automatic Control, Vol. AC-22, April 1977, pp. 173-179.
8. Safanov, M. G., and Athans, M., "Robustness and Computational Aspects of Nonlinear Stochastic Estimators and Regulators," IEEE Trans. on Automatic Control, Vol. AC-23, August 1978, pp. 718-725.
9. Athans, Michael and Falb, Peter L., Optimal Control, McGraw-Hill, NY.
10. Åström, K. J. and Eykhoff, P., "System Identification-a Survey," Automatica, Vol. 7, 1971, pp. 123-162.
11. Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, NY, 1970.
12. Bierman, G. J., Factorization Methods for Discrete Sequential Estimation, Academic Press, 1977.
13. Bierman, G. J., "Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study," Automatica Vol. 13, Jan. 1977, pp. 23-35.

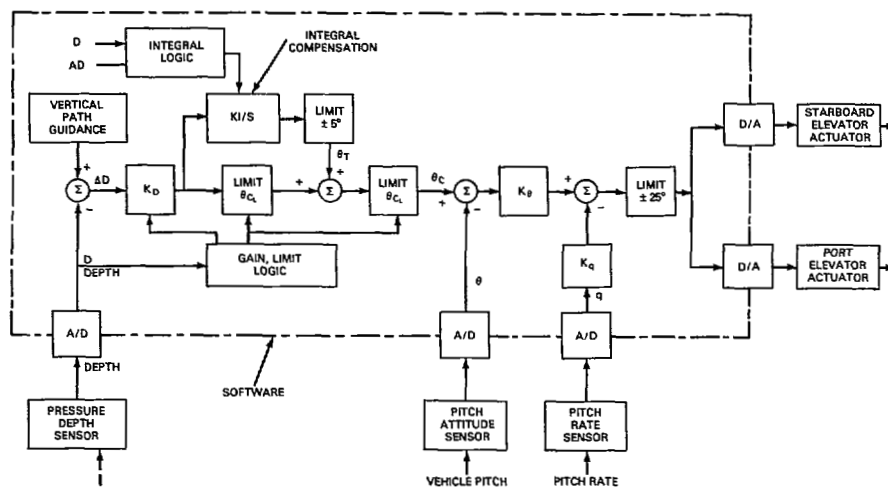


Fig. 6 Depth Control Mechanization Details

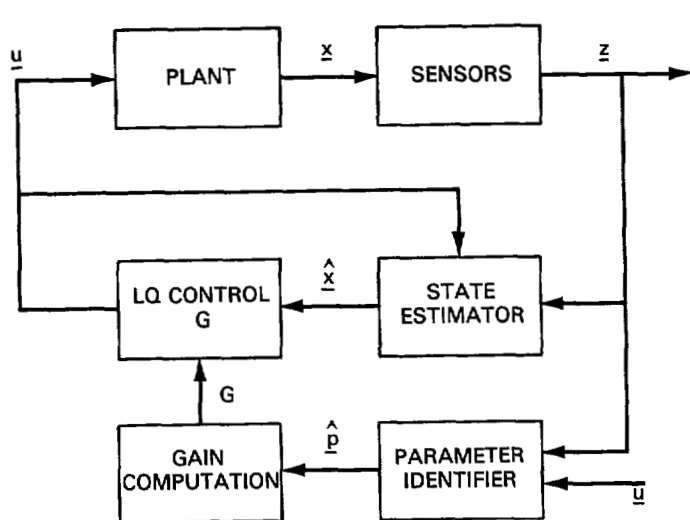


Fig. 7 Adaptive Control Structure

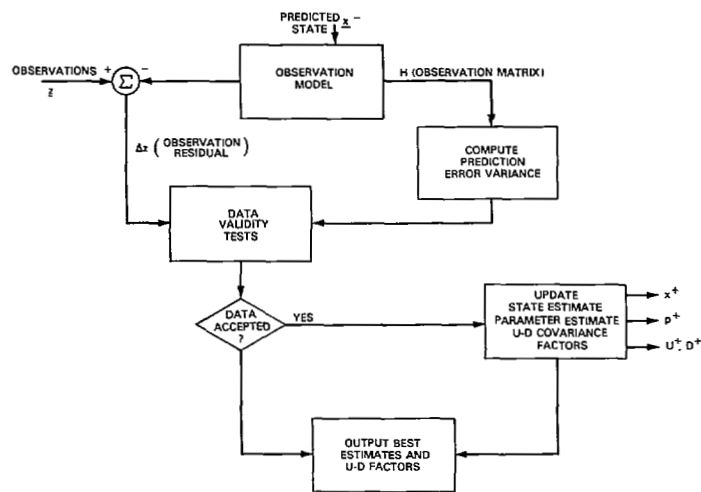


Fig. 8 Measurement Update

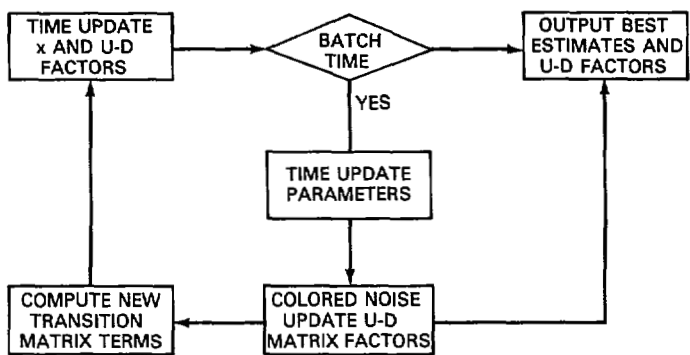


Fig. 9 Time Update

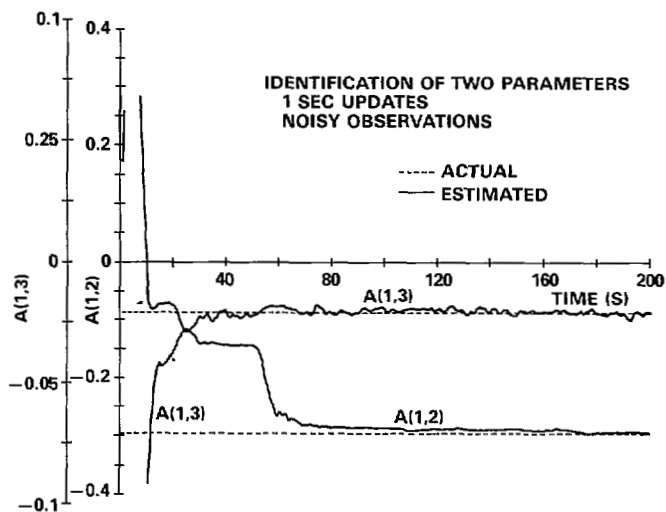


Fig. 10 EKF Parameter Identification from Simulated Test Data